УДК 004.056.5 (076.5)

Б. Я. Корнієнко, канд. техн. наук, доц.
С. О. Сулима, студ.

АНАЛІЗ ЗАХИЩЕНОСТІ SPANNING-TREE PROTOCOL

НАУ, кафедра комп`ютеризованих систем захисту інформації, e-mail: semen.sulyma@gmail.com

Стрімкий розвиток інформаційних технологій спричиняє появу дедалі більшої кількості вразливостей та атак, що використовують ці вразливості. Тому доводиться слідом (а, іноді, і випереджаючи події) розроблювати нові механізми захисту інформації, а також удосконалювати старі. Розглянуто Spanning-Tree Protocol – яскравий приклад того, як механізми безпеки іноді, закриваючи одну вразливість, створюють десять нових.

Extraordinary sweeping of IT – development causes vulnerabilities and, thereafter, attacks that use these vulnerabilities. That is why one must post factum or even in advance speed up invention of new information security systems as well as develop the old ones. The matter of article concerns Spanning-Tree Protocol – the vivid example of the case, when the cure of the vulnerability creates dozen of new "weak spots".

Вступ

Як відомо, існує чотири основні характеристики інформації – доступність, цілісність, спостережуваність і конфіденційність. В умовах стрімкого проникнення інформаційних технологій (ІТ) у наше життя критичність кожної з них надзвичайно зростає. Досить підрахувати ризики, пов'язані з простоєм великого підприємства через збій в ІТ або крадіжкою важливої інформації. Тому очевидно, що безпеці інформаційних технологій варто приділяти особливу увагу [1].

Постановка завдання

Розглянемо аспекти безпеки для побудови великих комутувальних мереж передавання даних. Слід зазначити, що грамотний дизайн мережевої архітектури, крім криптографічної безпеки, включає також механізми захисту від атак на доступність, що часом буває важливіше спостережуваності, конфіденційності й цілісності (така тенденція властива, зокрема, мережам магазинів роздрібної торгівлі, інтернет-бізнесу, логістичним центрам, де потрібна висока доступність для підтримки безперервності бізнесу). Одним з таких механізмів є Spanning-Tree Protocol (STP) – протокол зв'язаного дерева, що використовується у мережах передавання даних Ethernet. Основне призначення протоколу полягає в тому, щоб не допустити утворення петель у мережі з явною надлишковістю (рис.1). Надлишковість же, у свою чергу, підвищує загальну завантаженість мережі за рахунок наявності множинних зв'язків [1; 11; 13].

Теорія графів

Алгоритм Spanning-Tree реалізовано на основі теорії графів.

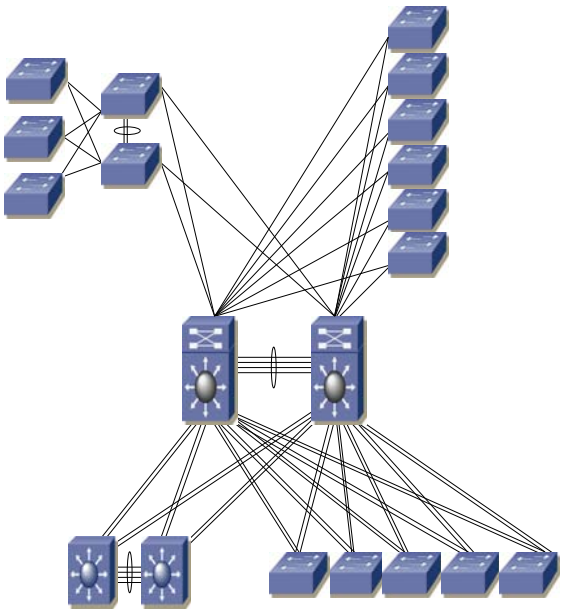

Рис. 1. Топологія комутувальної мережі передавання даних з надлишковістю

У загальному випадку є граф $G(V,E)$, де V – це множина вершин; E – множина їхніх попарних з'єднань (ребер) (рис. 2).

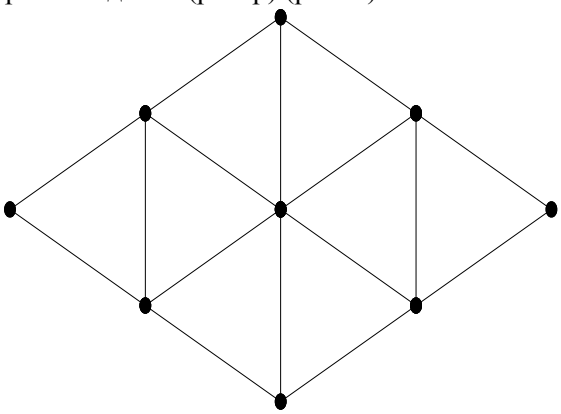

Рис. 2. Граф G

Кожне ребро характеризується ваговою функцією $w(u,v)$. Інакше кажучи, це вартість з'єднання. Завдання алгоритму Spanning-Tree зводяться до знаходження мінімального зв'язаного дерева $T, T \subset G$, для якого вартість сумарного з'єднання

$$w(T) = \sum_{(u,v)\in T} w(u,v) \qquad (1)$$

буде мінімальною (рис. 3) [7; 12]:

$w(T) = \sum_{(u,v)\in T} w(u,v) = 5+3+4+3+4+6+5+3 = 33 \cdot$

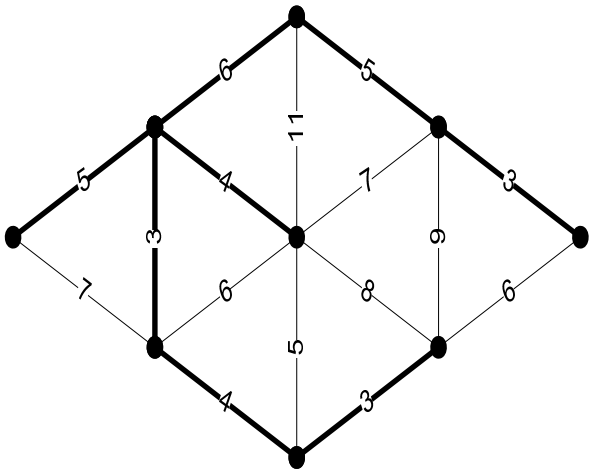

Рис. 3. Граф $T \subset G$ – мінімальне зв'язне дерево

Існує декілька способів знаходження мінімального зв'язаного дерева. В основі кожного з них лежить циклічний підграф A вихідного графу G. На самому початку граф G складається з n-вершин, не з'єднаних одна з одною. За кожної ітерації до підграфу A додається одне ребро. При цьому виконується одна важлива умова – $A \subset T$.

Алгоритм Борувки

Для кожної компоненти зв'язності вибирають «лідера» (або вершину). Після того як лідерів обрано, для кожної компоненти зв'язності знаходять безпечне ребро (мінімальну вартість з'єднання), що додається до підграфу A. Цей пошук відбувається доти, доки в підграфі A буде більше від однієї компоненти зв'язності (рис. 4).

Алгоритм Крускала

На кожному етапі вибирають ребро з найменшою вагою. Для кожного наступного ребра (перебирання іде за зростанням) перевіряється, чи не лежать кінці ребер в одній компоненті зв'язаності. Якщо так, то таке ребро не є безпечним і не додається до компоненти зв'язаності (рис. 5).

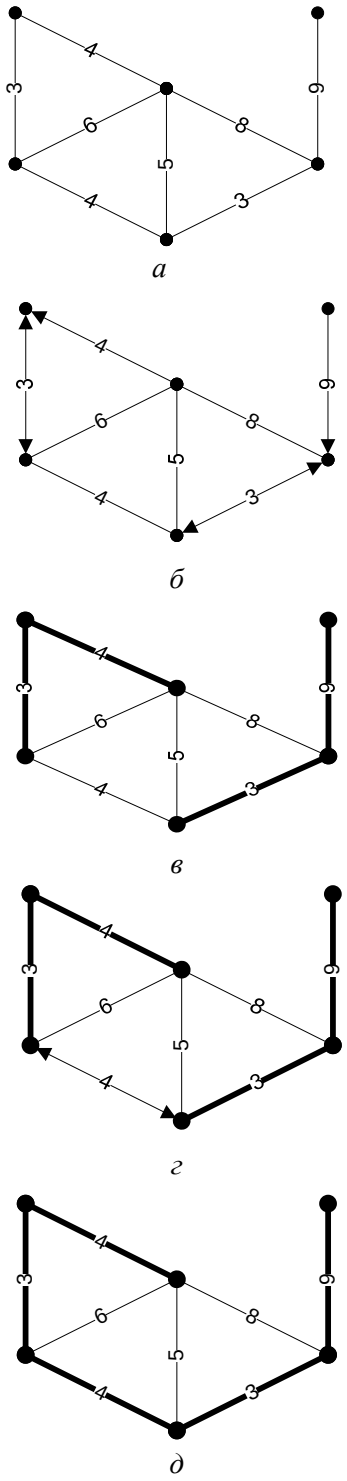

Рис. 4. Алгоритм Борувки: *а* – початковий стан. (Підграф A складається з n вершин і порожньої множини ребер); *б* – для кожної компоненти зв'язності (вершини) знаходимо безпечне ребро (позначено стрілками); *в* – додаємо ребра до підграфу A; *г* – для кожної компоненти зв'язаності знаходимо безпечні ребра (відмічені стрілками); *д* – додаємо ребра до підграфу A. (Мінімальне зв'язне дерево T побудовано.)

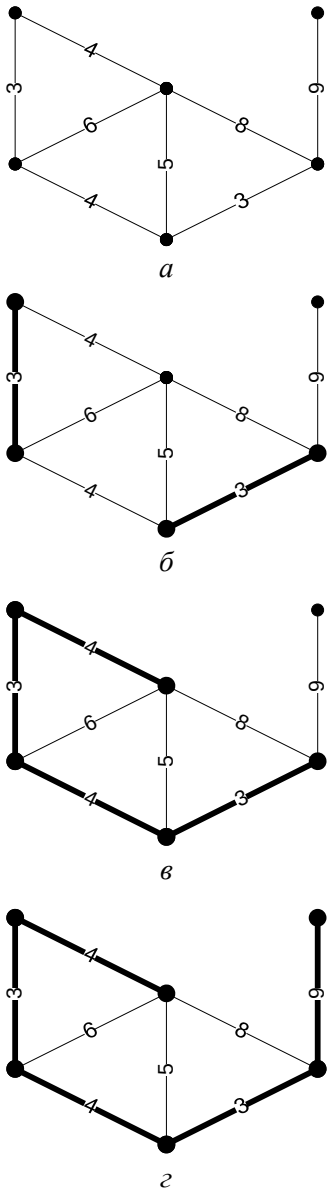

Рис. 5. Алгоритм Крускала: *а* – початковий стан. (Підграф A складається з n вершин і порожньої множини ребер); *б* – перебираємо ребра в порядку зростання: перші два ребра вартістю 3, додаємо їх до підграфу A; *в* – додаємо ребра вартістю 4 до підграфа A; *г* – додаємо ребро з вартістю 9 до підграфу A. (Кінці ребер 5, 6 і 8 лежать в одній компоненті зв'язаності)

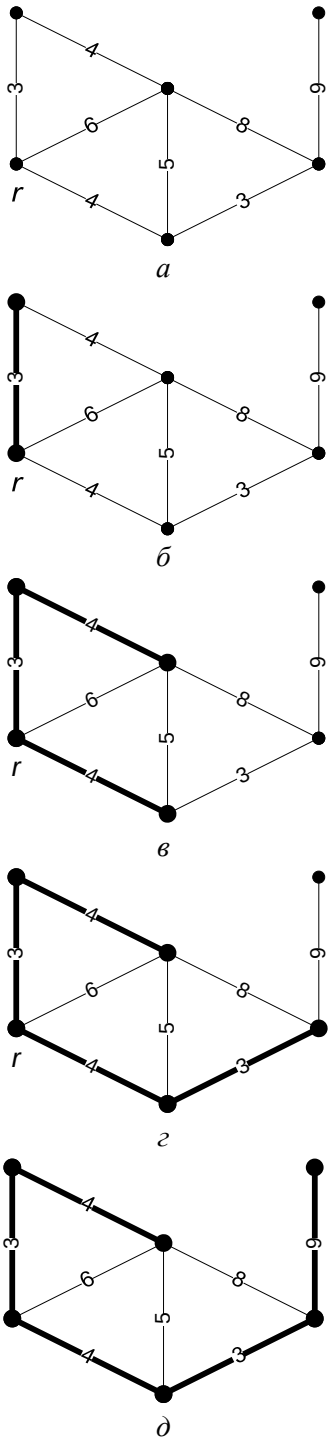

Рис. 6. Алгоритм Прима:*а* – початковий стан. (Підграф A складається з кореня r й порожньої множини ребер); *б* – *р*ебро вартістю 3, що з'єднує корінь r з іншими вершинами, є мінімальним; *в* – додаємо ребра вартістю 4 до підграфу A; *г* – додаємо ребро вартістю 3 до підграфу A; *д* – додаємо ребро з вартістю 9 до підграфу A. (Мінімальне зв'язне дерево T побудовано.)

Алгоритм Прима

У першу чергу вибирається корінь r – одна з вершин графу G. На кожному етапі до підграфу A додається ребро з найменшою вартістю (спочатку до кореня r) доти, доки підграф A не перетвориться в мінімальне зв'язне дерево T (рис. 6) [4; 6; 7].

Принцип роботи Spanning-Tree Protocol (STP)

Алгоритм Spanning-Tree являє собою мережу у вигляді графу, де комутатори (і мости) є вершинами, а мережеві сегменти – ребрами цього графу. Такий граф має одну кореневу вершину – кореневий комутатор, до якого розраховується найкоротший шлях від кожного комутатора. Такий найкоротший шлях і є мінімальним зв'язним деревом, побудову якого розглянуто раніше. Ваговою функцією є вартість мережевого сегмента. Цей підхід гарантує відсутність петель і раціональність проходження потоків даних у мережі [2].

Функціонування STP здійснюється за допомогою спеціальних кадрів Ethernet – BPDU (Bridge Protocol Data Unit), що являють собою мультикастові кадри Ethernet (з MAC адресою призначення – 01-80-C2-00-00-00). BPDU бувають двох видів: конфігураційний BPDU (далі c-BPDU) і TCN (Topology Change Notification); c-BPDU складається з набору полів, з яких цікавлять такі: ідентифікатор кореневого комутатора (Root Bridge Identifier RBID), відстань до кореня (Root Path Cost RPC), ідентифікатор комутатора (Bridge Identifier BID) (рис. 7).

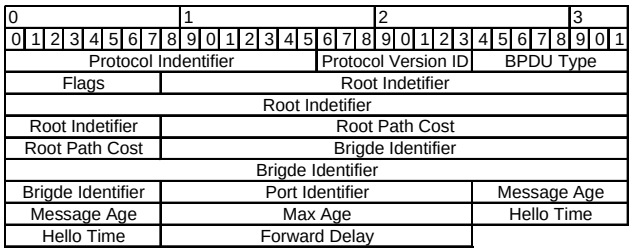

Рис. 7. Формат c-BPDU

Кореневий комутатор вибирається на основі ідентифікатора комутатора (BID), що складається з MAC адреси комутатора і його пріоритету, причому, чим нижчий пріоритет, тим більша ймовірність, що цей комутатор стане кореневим. За замовчуванням кожен комутатор у мережі має однаковий пріоритет – 32768, і, якщо не задати його вручну, то кореневим стане комутатор з найменшою MAC адресою. Перед вибиранням кожен комутатор вважає себе кореневим, розсилаючи при цьому c-BPDU зі значенням RBID, що дорівнює його власному значенню BID. Але як тільки він отримує c-BPDU від іншого комутатора, значення RBID якого менше від його власного, комутатор перестає вважати себе кореневим, і, відповідно, генерувати c-BPDU (він лише ретранслює c-BPDU, отримані від кореневого комутатора).

Кожен комутатор прораховує найкоротший шлях до кореневого комутатора на основі поля RPC c-BPDU, де зберігається сума вартостей кожного сегмента мережі до кореневого комутатора, і вартості власних портів. Порт, значення RPC через який мінімальне, стає кореневим. Порт, що не є кореневим, але входить у мінімальне зв'язне дерево, призначений. Всі інші порти блокуються комутатором. У такому стані заблокований порт може лише приймати й передавати c-BPDU, але ніяк не мережевий потік даних.

TCN може бути згенерованим будь-яким комутатором із STP (на відміну від c-BPDU) для повідомлення про зміну топології в мережі. При цьому відбувається очищення ARP таблиці кожного комутатора, на якому включено STP (для запобігання ситуації, коли пакети будуть передаватися на вже неіснуючий хост) [8; 10].

Вразливості STP

Як і безліч інших протоколів, STP не позбавлений недоліків. Більше того, будучи розробленим у 80 – 90-х роках XX ст., на сьогодні він являє собою серйозну загрозу для мережі поряд зі своїми перевагами. Так, наприклад, підвищена доступність мережі може дуже легко перетворитися в повну непрацездатність. В таблиці наведено можливі атаки, що використовують ті або інші вразливості STP [3; 1; 2].

Варто враховувати, що будь-яка атака, пов'язана з перевиборами кореневого комутатора, неминуче призводить до тимчасової втрати працездатності мережі. Як видно з таблиці, недоліків у STP більше, ніж переваг, тому для підвищення захищеності самого STP варто виконати ряд умов:

1. Використовувати замість STP RSTP (Rapid STP) [8], якщо є один VLAN, або MSTP (Multiple STP) [9], якщо є більше одного VLAN. У цих модифікацій STP у кілька разів прискорена конвергенція, тобто потрібно менше часу від початку виборів кореневого комутатора до передавання потоків даних. Також MSTP допоможе захиститися від атаки з об'єднанням дерев різних VLAN, тому що в реалізації з MSTP будується лише одне спільне зв'язне дерево (хоча цей параметр можна змінити).

2. Задати пріоритет на бажаному кореневому комутаторі,що дорівнює нулю, щоб знизити ймовірність того, що атакуючий виграє вибори (це не допоможе, якщо атакуючий вручну задасть MAC адресу свого комп'ютера, значення якої буде меншим, ніж у бажаного кореневого комутатора, у сумі з пріоритетом = 0) [11].

Атаки на STP

№ п/п	Назва атаки	Тип атаки	Опис
1	Постійні вибори кореня	Відмова в обслуговуванні (Denial of Service - Do)	Атакуючий «прослуховує» мережу й довідується про ідентифікатор кореневого комутатора (RBID). Далі генерує кадр BPDU, у якому підставляє значення RBID, що на одиницю менше від поточного значення RBID кореневого комутатора. Відбуваються перевибори кореня (під час яких мережа переходить у непрацездатний стан), які виграє атакуючий. Потім атакуючий ще й ще знижує значення RBID у BPDU. Як результат мережа повністю непрацездатна
2	Зникнення кореня	Відмова в обслуговуванні (Denial of Service - Do)	Метод, аналогічний першому, за винятком того, що атакуючий не «прослуховує» мережу, а відразу посилає BPDU з RBID = 0. Відбуваються перевибори кореневого комутатора, які виграє атакуючий. Потім корінь «зникає», що знову приводить до перевиборів. І так далі. Мережа також переходить у неробочий стан
3	Поділ мережі та фільтрація потоків даних	Відмова в обслуговуванні (Denial of Service - Do)	Атакуючий підключається до мережі комп'ютером із двома мережевими адаптерами. Потім генерує c-BPDU з мінімальним RBID і виграє вибори STP. Виходячи з теорії графів мінімальне зв'язне дерево обов'язково проходить через корінь, тобто через комп'ютер атакуючого. Атакуючий може проводити будь-які дії із транзитним потоком даних – розділити мережу на два незв'язані між собою сегменти, організувати фільтрацію потоків даних і т. ін.
4	Об'єднання дерев різних VLANів	Відмова в обслуговуванні (Denial of Service - Do)	Атакуючий підключається до мережі комп'ютером із двома мережевими адаптерами – кожним у різному VLAN, здійснюючи пересилання BPDU з одного VLAN в інший. У результаті дерева STP обох VLAN «побачать» один одного, що призведе до перевиборів кореня. Потім атакуючий відключається від мережі й знову відбуваються перевибори
5	Фільтрація BPDU	Відмова в обслуговуванні (Denial of Service - Do)	Атакуючий створює петлю в мережі й фільтрує на цій ділянці BPDU (щоб неможливо було відстежити петлю засобами STP), у результаті чого мережа через якийсь час стає непрацездатною
6	Очищення ARP таблиць комутаторів	Розвідка (reconnaissance)	Атакуючий посилає в мережу TCN BPDU, з одержанням якого кожен комутатор з STP очищає свою ARP таблицю, після чого переходить у режим концентратора, розсилаючи пакети в усі свої порти, поки ARP таблиця знову не наповниться. У цей момент атакуючий має доступ до даних у мережі [5]
7	«Людина посередині»	Атака доступу (access attack)	Атакуючий підключається до мережі комп'ютером із двома мережевими адаптерами. Потім генерує c-BPDU з мінімальним RBID і виграє вибори STP. У результаті весь потік даних перейде через комп'ютер атакуючого, котрий може робити над ними будь-які маніпуляції (читання, модифікацію, видалення й т.д.)

3. Запровадити механізми аутентифікації 802.1x на кожному порті кожного комутатора за допомогою сервера аутентифікації RADIUS.

4. Запровадити систему виявлення й запобігання вторгненням (IDS/IPS).

5. По можливості використати замість STP інші механізми резервування й підвищення доступу (наприклад, HSRP, LACP, динамічну маршрутизацію, власні розробки виробників [13] й ін.).

Висновки

Таким чином, з упевненістю можна говорити про те, що STP – це аж ніяк не панацея для вирішення проблеми надлишковості в мережах передавання даних. Його використання вимагає, у першу чергу, глибокого розуміння, навіщо він потрібний і чи потрібний він взагалі, оскільки ціна помилки вибору рішення може бути дуже високою. Що ж стосується спроб різних виробників підвищити безпеку STP, то вони хоч і вирішують проблему, але дуже локально. Насамперед потрібно впроваджувати механізми аутентифікації кадрів BPDU (з використанням хешування) [3]. Цей підхід разом із шифруванням інформації BPDU дозволить істотно підвищити безпеку STP. Поки ж необхідно або задовольнятися існуючими реалізаціями STP, або впроваджувати альтернативні рішення підвищення доступності.

Література

1. *Sean Convery*. Network Security Architectures. – Cisco Press, 2005. – 792 p.
2. *David Hucaby*. CCNP Self-Study: CCNP BCMSN Exam Certification Guide, Third Edition. – Cisco Press, 2005. – 624 p.
3. *Oleg Artemjev, Vladislav Myasnyankin*. Are those loops on your network neck secure? – M.: Open Systems, 2002. – 13 p.
4. *Gengui Zhou, Mitsuo Gen*. Genetic Algorithm Approach on Multi-criteria Minimum Spanning Tree Problem. – Department of Industrial and Systems Engineering Ashikaga Institute of Technology, Ashikaga, Japan, 1997. – 17 p.
5. *Sean Whalen, Matt Bishop*. Layer 2 Authentication. – 2005. – 12 p.
6. *Sean Odom, Hanson Nottingham*. Cisco Switching Black Book. – Coriolis Group, 2001. – 395 p.
7. *Глеб Рыбаков*. Построение минимального остовного дерева (алгоритмы Крускала, Прима, Борувки). – М.: Open Systems, 2005. – 14 с.
8. *Cisco* Systems, Understanding Rapid Spanning Tree Protocol (802.1w) (Document ID: 24062). – 2006. – 14 p.
9. *Cisco* Systems, Understanding Multiple Spanning Tree Protocol (802.1s) (Document ID: 24248). – 2005. – 14 p.
10. *Dave Hucaby, Steve McQuerry*. Cisco Field Manual: Catalyst Switch Configuration. – Cisco Press, 2002. – 560 с.
11. *KnowledgeNet*, Security+ (Student's guide version 1.0). – KnowledgeNet, 2003. –471 с.
12. *Micha Pióro, Deepankar Medhi*. Routing, Flow, and Capacity Design In Communication and Computer Networks. – Morgan Kaufmann, San Francisco, 2004. – 794 p.
13. *Paola Flocchini, Alessandro Roncato, Nicola Santoro*. Computing on anonymous networks with sense of direction. – School of Information Technology and Engineering, University of Ottawa, 2002. – 25 p.

Стаття надійшла до редакції 22.11. 06.